IN THE UNITED STATES DISTRICT COURT

FOR THE NORTHERN DISTRICT OF CALIFORNIA

NETWORK APPLIANCE INC,

Plaintiff,

v.

SUN MICROSYSTEMS INC,

Defendant.

No. C-07-06053 EDL

ORDER GRANTING SUN MICROSYSTEMS INC.'S MOTION NO. 1 FOR SUMMARY JUDGMENT OF NON-INFRINGEMENT OF U.S. PATENT NO. 6,892,211

I. INTRODUCTION

On September 5, 2007, Network Appliance, Inc. ("NetApp") filed its Complaint, alleging that Sun Microsystems, Inc. ("Sun") infringed and is infringing, directly and indirectly under 35 U.S.C. § 271, certain of its patents, by making, using, selling, or offering for sale certain data processing systems and related software. NetApp seeks a declaratory judgment that certain patents owned by Sun are each not infringed, are invalid and/or are unenforceable, as well as a permanent injunction and damages. On October 25, 2007, Sun filed an Answer and Counterclaim, denying the material allegations of NetApp's Complaint and asserting a number of affirmative defenses and counterclaims. Sun denies infringing any of the NetApp Patents, including the patent at issue in this motion (U.S. Patent Number 6,892,211 (the "'211 patent")) and alleges that NetApp infringes a number of its patents instead. On September 10, 2008, this Court issued an Order Construing Claims (the "9/10/08 Order") in which it construed fourteen disputed terms and/or phrases contained in various claims in the seven patents at issue between the parties, including two terms contained in the '211 patent. The parties subsequently conducted discovery, and each party has filed two summary judgment motions in the above-captioned 07-6053 case.

United States District Court
For the Northern District of California

On August 3, 2009, Sun filed Motion No. 1 For Summary Judgment Of Non-Infringement Of U.S. Patent No. 6,892,211 ("Motion No. 1") on the basis that its allegedly infringing products do not contain an "on-disk root inode" as that term has been construed by the Court. Motion No. 1 was fully briefed, and a hearing was held on September 23, 2009. Having considered the record in this case and the parties' statements at oral argument, and for the reasons set forth below, the Court hereby GRANTS Sun's Motion No. 1 For Summary Judgment Of Non-Infringement of the '211 patent.

II. LEGAL STANDARD

A. Summary Judgment

Summary judgment shall be granted if "the pleadings, discovery and disclosure materials on file, and any affidavits show that there is no genuine issue as to any material fact and that the movant is entitled to judgment as a matter of law." Fed. R. Civ. Pro. 56(c). Material facts are those which may affect the outcome of the case. See Anderson v. Liberty Lobby, Inc., 477 U.S. 242, 248 (1986). A dispute as to a material fact is genuine if there is sufficient evidence for a reasonable jury to return a verdict for the nonmoving party. Id. The court must view the facts in the light most favorable to the non-moving party and give it the benefit of all reasonable inferences to be drawn from those facts. Matsushita Elec. Indus. Co. v. Zenith Radio Corp., 475 U.S. 574, 587 (1986). The court must not weigh the evidence or determine the truth of the matter, but only determine whether there is a genuine issue for trial. Balint v. Carson City, 180 F.3d 1047, 1054 (9th Cir. 1999).

A party seeking summary judgment bears the initial burden of informing the court of the basis for its motion, and of identifying those portions of the pleadings and discovery responses that demonstrate the absence of a genuine issue of material fact. Celotex Corp. v. Catrett, 477 U.S. 317, 323 (1986). Where the moving party will have the burden of proof at trial, it must affirmatively demonstrate that no reasonable trier of fact could find other than for the moving party. On an issue where the nonmoving party will bear the burden of proof at trial, the moving party can prevail merely by pointing out to the district court that there is an absence of evidence to support the nonmoving party's case. Id. If the moving party meets its initial burden, the opposing party "may not rely merely on allegations or denials in its own pleading;" rather, it must set forth "specific facts

showing a genuine issue for trial." See Fed. R. Civ. P. 56(e)(2); Anderson, 477 U.S. at 250. If the nonmoving party fails to show that there is a genuine issue for trial, "the moving party is entitled to judgment as a matter of law." Celotex, 477 U.S. at 323.

B. Patent Infringement

"To prove infringement, the patentee must show that the accused device meets each claim limitation either literally or under the doctrine of equivalents." Catalina Mktg. Int'l v. Coolsavings.com, Inc., 289 F.3d 801, 812 (Fed. Cir. 2002). A determination of infringement, whether literal or under the doctrine of equivalents, is a question of fact. Id. "Literal infringement requires the patentee to prove that the accused device contains each limitation of the asserted claim." Id. "Summary judgment of no literal infringement is proper when, construing the facts in a manner most favorable to the nonmovant, no reasonable jury could find that the accused system meets every limitation recited in the properly construed claims." Id. Where the parties do not dispute any relevant facts regarding the accused product, but disagree over possible claim interpretations, the question of literal infringement collapses into claim construction and is amenable to summary judgment. General Mills, Inc. v. Hunt-Wesson, Inc., 103 F.3d 978, 983 (Fed. Cir. 1997); cf. Int'l Rectifier Corp. v. IXYS Corp., 361 F.3d 1363, 1375 (Fed. Cir. 2004) (distinguishing General Mills on the basis that only the structure of the accused devices had been stipulated to, not the disputed factual determination of whether the device met the claims as construed, but not addressing the scenario in which no reasonable juror could find that a certain claim limitation was met).

In MyMail Ltd. v. America Online, Inc., 476 F.3d 1372, 1378 (Fed. Cir. 2007), the Federal Circuit reviewed a District Court order granting summary judgment of non-infringement. Because there were no material factual disputes as to the operation of the accused systems, and because the disagreements concerned whether the defendants' systems performed "authentication" as defined by the patent and construed by the district court, the Federal Circuit found that the issue reduced to a question of claim interpretation and affirmed summary judgment. See id. (noting that the accused product did not satisfy the authentication requirement as it did not validate the user's ID and password, as required by the patent's authentication process). These cases teach that the Court cannot leave it to the jury to decide the proper scope of the patent claim terms. 02 Micro Int'l Ltd. v.

Beyond Innovation Tech. Co. Ltd., 521 F.3d 1351, 1360 (Fed. Cir. 2008) ("When the parties raise an actual dispute regarding the proper scope of the[] claims, the court, not the jury, must resolve the dispute.").

"Infringement under the doctrine of equivalents requires the patentee to prove that the accused device contains an equivalent for each limitation not literally satisfied." Id. The Court may not apply the doctrine of equivalents so as to vitiate a claim limitation. Warner-Jenkinson, 520 U.S. at 29, 39 n.8. The Federal Circuit articulates the test for equivalence in two different ways. See Voda v. Cordis Corp., 536 F.3d 1311, 1326 (Fed. Cir. 2008). Under the insubstantial differences test, "[a]n element in the accused device is equivalent to a claim limitation if the only differences between the two are insubstantial." Honeywell Int'l Inc. v. Hamilton Sundstrand Corp., 370 F.3d 1131, 1139 (Fed.Cir.2004); Warner-Jenkinson Co. v. Hilton Davis Chem. Co., 520 U.S. 17, 40 (1997)). Alternatively, under the function-way-result test, an element in the accused device is equivalent to a claim limitation if it "performs substantially the same function in substantially the same way to obtain substantially the same result." Schoell v. Regal Marine Indus., Inc., 247 F.3d 1202, 1209-10 (Fed. Cir. 2001). "Where the evidence is such that no reasonable jury could determine two elements to be equivalent," summary judgment of non-infringement under the doctrine of equivalents is proper. Warner-Jenkinson, 520 U.S. at 39 n.8. Summary judgment has been rejected because of conflicting expert testimony on the application of the function-way-result test. Crown Packaging Tech., Inc. v. Rexam Bev. Can Co., 559 F.3d 1308, 1315 (Fed. Cir. 2009) (holding that conflicting expert evidence regarding function establishes material issue of fact).

III. ANALYSIS

A. The Patent At Issue

The '211 patent is a continuation of NetApp's U.S. Patent No. 5,819,292 and is similarly directed to a method for keeping a file system in a consistent state. The '292 patent describes a method for maintaining consistent states of a file system, and for creating snapshots that are read-only copies of the file system. '292 Patent Abstract. The invention uses a write anywhere file-system layout ("WAFL"). The file system progresses from one self-consistent state to another self-consistent state. The set of self-consistent blocks on disk that is rooted by a root inode is referred to

as a consistency point. The root inode is stored in a file system information structure. Id. In the detailed description of the invention, the invention is described as a system that utilizes storage blocks on the disk, and disk drives provide the storage space for the file system and maintain the structure and content of the file system. The system also uses an in-memory "buffer" to hold changes to the file system prior to storing them on the disk. The data blocks are organized, described, and pointed to by inode blocks. Id. at 5-6. The inode files themselves are pointed to by a "root inode." Id. at 9:34-35. When changes to the file system occur during use, WAFL writes new or modified data to unallocated blocks on disk so that it never overwrites existing data. Id. at 12:2-4. The file system in this patent can also retain a copy of older data in a prior consistent state in a read-only form called a snapshot. Id. at 4:20-21.

The '211 patent shares the same specification as the '292 patent, and contains more detailed mechanics of a file system that progresses from one consistent state to another. The system utilizes a root inode that contains pointers that directly point to a metadata file known as the inode file that contains the inodes of all of the other files in the file system. Homrig Decl, Ex. 2 ('211 Patent) at 9:25-33. Each file may be broken up into data blocks of 4KB and is described by a corresponding inode, that contains pointers that point to data blocks or to a set of indirect blocks that point to data blocks. Id. at 5-6. It is important for a file system to maintain a consistent state on disk to which the system can revert in case a failure occurs while the file system is being changed. Id. at 1:24-26, 13:63-14:6.

The '211 patent includes three independent claims – claims 1, 9, and 17 – each of which is asserted against Sun. NetApp also asserts dependent claims 2, 3, 10, 11, 18, and 19. The three independent claims are similar, and Claim 9 is for:

> a device comprising:
> a processor;
> a memory; and
> a storage system including one or more hard disks;
> wherein said memory and said storage system store a file system; and
> wherein said memory also stores information including instructions executable by said processor to maintain said file system, the instructions including steps of (a) maintaining an **on-disk root inode** on said storage system, said on-disk root inode pointing directly and indirectly to a first set of blocks on said storage system that store a first consistent state of said file system, and (b) maintaining an incore root inode in said memory, said incore root inode pointing directly and indirectly to buffers in said memory and a second set of blocks on said storage system, said buffers and said second set of blocks storing data and meta-data for

> a second consistent state of said file system, said second set of blocks including at least some blocks in said first set of blocks, with changes between said first consistent state and said second consistent state being stored in said buffers and in ones of said second set of blocks not pointed to by said on-disk inode.

Id. at 24:39-62 (emphasis added).

The Court has construed the term "on-disk root inode"as "the index node data structure stored in a fixed location on disk that roots a set of self-consistent blocks on the storage system that comprise the file system." Sept. 10, 2008 Claim Construction Order (hereinafter "9/10/08 Order") at 45. The Court observed:

> Multiple portions of the patent specification teach that the root inode roots a set of blocks that comprise the file system in a consistent state. See, e.g., '211 Patent at 11:20-27, 11:65-67. The specification teaches that the "root inode 1510B of a file system is kept in a fixed location on disk so that it can be located during booting of the file system." Id. at 10:59-61; see also id. at 9:33-35 (root inode is kept on a fixed location on disk referred to as the file system information block described below). The root inode is the inode that locates the inode file, which stores inodes for the other files in the file system. Id. at 9:32-33. The invention teaches that the on-disk root inode is stored in a fixed location on the fsinfo block to enable the location of the inode files, which are written to any available locations on disk and may move around, in accordance with the "write anywhere" nature of the file system. Id. at 9:19-24. See also Brandt Decl. ¶¶ 127-29. The abstract itself notes that the "set of self-consistent blocks on disk is rooted by a root inode." The use of the terms "root" and "rooted" suggest that the root inode is not rootless but rather is fixed or stored at some set location, although their primary meaning is serving as the base of the tree. '211 Patent at 11:24-25. These portions of the specification address the on-disk root inode.
> ...
>
> NetApp also argues that even an on-disk root inode is not necessarily stored at a fixed location. NetApp largely relies on Dr. Ganger's declaration, but Dr. Ganger does not point to the specification in making his observations, so this extrinsic evidence is not particularly persuasive. See Ganger Decl. ¶¶ 79-80. Dr. Ganger notes that there are other mechanisms for ensuring that the root inode can be located. For example, the file system could store a pointer to the root inode in a fixed location. But a pointer that points to other inodes in non-fixed locations is really just another way of describing a root inode. Dr. Ganger also notes that a file system could have a set of predetermined locations that might hold the root inode. However, Dr. Ganger has not demonstrated how the specification might contemplate such an embodiment, and the specification itself repeatedly refers to a "fixed location," as discussed above. However, if the set were small enough that the root inode could be readily located, it might constitute a fixed location (or at least its equivalent), as Sun's expert acknowledged at the hearing.

Id. at 43-44.

B. NetApp's Infringement Contentions and Sun's Motion

Sun's ZFS is a general use file system technology that is incorporated into Sun's open-source

OpenSolaris operating system and its counterpart, Solaris 10 OS. ZFS operates as a storage pool that supports both file systems and other types of datasets. Multiple file systems can exist simultaneously within a common storage pool managed by ZFS. McKusick Decl. ¶ 6; Williamson Decl., Ex. 2 (ZFS Specification) at 5. ZFS is implemented in software as a component of Sun's OpenSolaris and Solaris 10 operating systems. McKusick Decl. ¶ 7. Those operating systems are licensed for use in non-Sun products and are also sold within Sun's own hardware products. Shapiro Decl. ¶ 7. Many of Sun hardware products can use one of several different operating systems. Certain products in which Solaris 10 OS is the installed operating system use the UFS file system as a default, rather than ZFS, although the customer can later configure its system to run ZFS in lieu of UFS. Id. ¶¶ 8-10. The Amber Road series of Sun products are pre-installed with software that runs ZFS only and cannot be reconfigured to run alternate files systems. Id. ¶ 11.

NetApp contends that the uberblock feature of ZFS is the "on-disk root inode" of the '211 patent. See, e.g., Williamson Decl., Ex. 3 (Ganger Report) at ¶¶ 39, 76, 115; Ex. 1 (NetApp Patent Local Rule 3-6 Disclosure) at 2, 3, 5. The ZFS uberblock is located within a 256 KB structure called a "vdev label." It is undisputed that each disk used by ZFS has four vdev labels, two at the "front" of each disk and two at the "back" of each disk. Sun Mot. at 9; NetApp Opp. at 11. The locations of the vdev labels are fixed when each disk is added to the pool. Williamson Decl., Ex. 2 at 7; see also Homrig Decl., Ex. 29 (Moore Depo Tr.) at 165:5-9 (contrasting MOS block locations as not being fixed, because "they can literally be anywhere, compared to the vdev labels, which can't be anywhere"); Ex. 1 (McKusick Depo Tr.) at 39 (discussing vdev label locations and noting that the four locations "are known locations and one can certainly say if it is a known location that it is a fixed location."). The parties do not dispute that the vdev labels are in fixed locations.

Illustration 3 from section 1.3 of the ZFS specification, reproduced on page 10 of Sun's opening brief, depicts the four parts of a vdev label. A vdev label includes an array of 128 individual 1K sized uberblock locations, totaling 128 KB of space. Williamson Decl., Ex. 2 at 8. An individual uberblock is written to one of the 128 uberblock locations. Id.; see also Ex. 9 (Ganger Depo) at 24, 69. An uberblock is "the portion of the label containing information necessary to access the contents of the pool" and "only one uberblock in the pool is active at any point in time."

United States District Court
For the Northern District of California

Williamson Decl., Ex. 2 at 12. The uberblock with the highest transaction group number and valid SHA-256 checksum (the most recent, valid uberblock) is the active uberblock. Id. "To ensure constant access to the active uberblock, the active uberblock is never overwritten. Instead, all updates to an uberblock are done by writing a modified uberblock to another element of the uberblock array." Id. at 12-13. In other words, the new uberblock for the pool is written to one of the 128 uberblock locations in the vdev labels other than the location to which the immediately prior uberblock was written. McKusick Decl. ¶¶ 9-10; Williamson Decl., Ex. 9 (Ganger Depo) at 27-28, 30-31 (noting that a new instance of the uberblock is stored each time for each new transaction that is synched). Over time, new uberblocks are written in a round-robin fashion across the 128 locations in the uberblock array. Williamson Decl., Ex. 2 at 13.

Sun moves for summary judgment of noninfringement of the '211 patent on the basis that Sun's products do not have an "on-disk root inode" because the "uberblock" identified by NetApp as the on-disk root inode is not stored in a "fixed location" on disk. Sun argues that the uberblock changes locations each time a transaction group is synched, and when the system needs to find the uberblock, it must search the disks in the storage pool, all vdev labels on each disk, and, within each vdev label, all of the 128 uberblock structures in the uberblock array in which the active uberblock could be located. Sun argues that this search could extend to hundreds or thousands of locations, and that no reasonable juror could find that the uberblock is stored in a "fixed location." NetApp counters that the uberblock is stored in the vdev label and uberblock array and that these are located are in a fixed location, which meets the Court's construction of the term.

The parties are in agreement as to how the ZFS uberblock is designed and operates. The parties agree that four "vdev label" structures on each disk are determined when each disk is added to the storage pool, and are therefore on fixed locations on disk. Sun Mot. at 9; NetApp Opp. at 10-15; Sun Reply at 2; see also Homrig Decl., Ex. 28 (Bonwick Depo. Tr.) at 20 (each vdev label is known at the offset); Ex. 29 (Moore Depo. Tr.) at 161 (conceding that locations of vdev labels can be calculated and "you could call that fixed"); Ex. 30 (Maybee Depo.) at 106 (noting that vdev label is at specific location on the disk). It is also undisputed that each vdev label includes a fixed uberblock array consisting of 128 1 KB locations in which an uberblock may be written. Sun Mot. at 9-10;

NetApp Opp. at 12-14; Sun Reply at 2-3; see also Homrig Decl., Ex. 1 (McKusick Depo. Tr.) at 41 (testifying during deposition that the uberblock array is in a fixed location relative to the beginning of a vdev label). NetApp also does not provide evidence to dispute that only one uberblock in the storage pool is active at a time, and that when a new active uberblock is written, it is written to one of the 128 locations other than the location where the prior uberblock was written. Sun Mot. at 10-11; NetApp Opp. at 10-18; Sun Reply at 3. Therefore the only issue before the Court is whether alternating between a predetermined set of at least 128 (and potentially, as discussed below, thousands of possible locations in a multi-disk system) which are collectively in a fixed location on disk can meet the Court's construction of the on-disk root inode limitation.

Sun argues that this issue is a question of law for the Court to decide on summary judgment; NetApp contends that if reasonable minds could differ then it is a fact question for the jury. To answer this question, the Court need not weigh conflicting evidence on which reasonable minds might disagree because the parties are in agreement as to the relevant components and structure of the ZFS system. Resolution of the motion requires only an application of the claim terms to the undisputed aspects of the accused technology, and the Court can decide this issue as a matter of law. <u>See General Mills, Inc. v. Hunt-Wesson, Inc.</u>, 103 F.3d 978, 983 (Fed. Cir. 1997).

C. Alternating Between Hundreds Of Possible Locations Does Not Meet The Court's Construction of A "Fixed" Location

Based on the undisputed facts, the ZFS active uberblock is not stored in a fixed location on disk as required by the Court's construction of "on-disk root inode." First, Sun argues that the active uberblock is specifically designed <u>not</u> to be in a fixed location, as it constantly changes location, moving across an array of 128 locations (the "uberblock array") in each of the four vdev labels on the disk. <u>See</u> McKusick Decl. ¶ 9; Bonwick Decl. ¶ 9. For the first vdev label on each disk, the uberblock array is located at the fixed location of 128 KB to 255 KB from block zero of the disk, and the other three uberblock arrays are stored in known locations relative to the beginning of each vdev label. Homrig Decl., Ex. 1 at 41. The question is whether or not the fact that the uberblock array is fixed, even though the uberblock itself moves within this array, satisfies the claim limitation.

NetApp's expert, Dr. Ganger, opined that one who is of ordinary skill in the art would

recognize that an on-disk root inode is stored in a fixed location if it can be located anywhere within 128 kilobytes worth of disk space, so long as that on-disk root inode were stored within a "structure specifically defined as a fixed location." Id., Ex. 33 (Ganger Depo. Tr.) at 54 ("I think if there was a structure specifically defined as a fixed location and the on-disk root inode were stored within that structure, right, within that fixed location structure, that yes, one of who [sic] is of ordinary skill in the art would recognize that as the on-disk root inode is stored in a fixed location.") NetApp therefore maintains that summary judgment is inappropriate because a reasonable juror could conclude that ZFS has an on-disk root inode which is stored in a fixed location on disk (i.e., a vdev label), as required by the Court's claim construction.

As discussed below, the meaning of "fixed" is an issue of law for the Court to determine, and the Court concludes that as a matter of law it is not so broad as to encompass hundreds of different locations, even if those locations are within a structure that itself could be seen as "fixed."

1. There Are At Least 512 Possible Locations for the Active Uberblock in a Single Disk System, and More In Multi-Disk Systems

According to Sun, the system must search all 128 locations on the uberblock array on each of four vdev labels on a disk when it needs to find the active uberblock, which means that the system must search 512 locations on a single disk to locate the active uberblock. See Williamson Decl., Ex. 9 (Ganger Depo.) at 32-34 (during cold boot ZFS must scan all four vdev labels in order to find uberblock and that each of four labels, which are meant to be replicas of each other, have 128 locations, and that in a single disk cold boot scenario, 512 locations must be scanned to find valid uberblock); McKusick Decl. ¶ 13; Bonwick Decl. ¶ 8. NetApp argues that there is a factual dispute as to whether 512, rather than 128, is the correct number of possible locations on each disk where the active uberblock could be stored. In support of its argument, NetApp relies on Dr. Ganger's testimony that the four vdev labels are intended to be replicas of each other. Williamson Decl., Ex. 9 (Ganger Depo.) at 32. Sun concedes that the location of the uberblock in each of the four vdev labels on a given disk should be the same during normal operation of a storage system. Sun Reply at 3. However, at oral argument, neither party disputed the fact that a system always searches all of the disks in a pool, all four of the vdev labels on each disk, and all 128 uberblock locations in each vdev

label to find the most recent valid uberblock. Sun Mot. at 12; NetApp Opp. at 12; Reply at 4. Therefore it is undisputed that in a single disk system, the system must search 512 locations to find the active uberblock, and more in a multi-disk system. See Williamson Decl., Ex. 9 (Ganger Depo) at 32-34, 40-41 (stating that in a one disk system, ZFS searches all 512 potential locations to find the active uberblock, even if they are replicas, and in a multi-disk system, ZFS must search thousands of potential locations). Additionally, even assuming that there were "only" 128 possible locations (which the undisputed facts show is not the case) that number is still too large to meet the definition of "fixed." Arguably, there might be some dispute of infringement, at least the doctrine of equivalents, if there were only two, or maybe four, possible locations, but the Court need not rule on that question as the issue is not before it.

2. The Term "Fixed" Is Not As Elastic As NetApp Contends

When the Court construed the term "on-disk root inode," the dispute focused on whether or not the root inode was fixed, rather than parsing the parameters of what is fixed and what is not. However, the plain meaning of the term "fixed" is the antithesis of what NetApp argues. While there may conceivably be other situations that would present a closer question, such as being located in one of only two possible places, something that can reside in hundreds (or even thousands) of different locations is not fixed.

Both parties argue that the Court's claim construction order and the experts' positions during claim construction support their arguments as to whether or not the root inode is in a "fixed" location. For example, NetApp's expert Dr. Ganger stated at the claim construction hearing that, "if there are multiple locations where it might be, and it moves over time from one of them to another, its location is not fixed. It might be a predetermined set, but it – clearly the location is not fixed, if it can be here, and then later it's here instead . . ." Williamson Decl., Ex. 8 at 83-84 (further stating "fixed means it does not move").[1] Sun's expert Dr. Brandt made a somewhat equivocal statement

---

[1]In a related context, NetApp argues that Dr. Ganger's statement that fixed means it does not move has no applicability here, because the active uberblock in ZFS is not moving in the sense that the previous version of the active uberblock stays where it is. Rather, a new instance of the active uberblock is stored in the next slot of the uberblock array. See Homrig Decl., Ex. 33 (Ganger Depo.) at 25 (does not understand the uberblock to move from one of 128 possible locations to another as a new transaction is committed), 26-28 (because there is a new instance of uberblock stored each time, he would not

that if something is in a set of predetermined locations, he was "not sure that's significantly different from 'a fixed location.'" Id. at 83 (also stating that "[i]f it's in a set of fixed locations, it's in a fixed location and a second fixed location, and a third . . ."). However, Dr. Ganger correctly pointed out that the patent contemplated only two, fixed, predetermined locations and correctly distinguished between the two terms, noting that"fixed" means "it doesn't move," and "predetermined" means "you know ahead of time where to look." Id. at 84. Indeed, the distinction between fixed and predetermined is established as a matter of plain English language.

NetApp also argues that when the Court construed the '211 patent, it concluded that the root inode was in a fixed location on disk – specifically the file system information or "fsinfo block." 9/10/08 Order at 40. The root inode could be anywhere within the fsinfo block, which itself is fixed, as there is no specific block address assigned to the root inode within the larger block. '211 patent at 10:58-11:5, Fig. 15. NetApp equates this arrangement to the ZFS system, which the parties agree has four vdev labels in fixed locations, two at the front of each disk and two at the back. Williamson Decl., Ex. 2 (ZFS Specification) at 7; Homrig Decl., Ex. 1 (McKusick Depo. Tr.) at 38-39. ZFS can find the vdev labels at the front of the disk, because those have a fixed block number, which can be used to calculate the two labels at the back of each disk, by counting back from the end and by determining how large the disk is. Id. at 42. Other blocks in ZFS, in contrast, can be "anywhere." Id., Ex. 29 (Moore Depo. Tr.) at 165. To the extent that NetApp relies on this Court's prior observation in the claim construction order that "the root inode is in a fixed location on disk, which is referred to as the file system information block," the Court made this statement in connection with the '292 patent, not the '211 patent, and in the different context of construing the "file system information structure" term. 9/10/08 Order at 41. The issue of the definition of the term "fixed" was not before the Court. Further, the construction of the "file system information structure" term cannot broaden the construction of the "on-disk root inode" term, and Dr. Ganger indicated that the "on disk root inode" construction should be incorporated into the "file system information structure"

---

characterize uberblock as moving). However, even if the active uberblock does not move and is in one location at any given time, the location of the active uberblock is still constantly changing within the array of the disk as the previous one is superceded and a new one in a different location replaces it. To find otherwise would rob the term "fixed" of any meaning whatsoever, because any moving thing would be "fixed" in time at any given instant.

construction, not vice versa. Williamson Reply Decl., Ex. F (Ganger Depo) at 122, 124. Finally, NetApp has made no showing that the fixed location in the fsinfo block is anywhere near as elastic as having 128 or 512 (or thousands of) possible locations such that it could be equated to the ZFS uberblock. In the '211 patent, Figure 15 shows that things other than the root inode are located in the fsinfo block. Specifically, there is miscellaneous data there, which need not even be contained in the fsinfo block. Homrig Decl., Ex. 2 ('211 Patent) at 11:1-3. There is no teaching that the root inode can exist in multiple locations in the fsinfo block. Therefore, this argument is not persuasive.

NetApp also focuses on another portion of the claim construction order which it claims indicates that a "small set of locations" might constitute a fixed location:

> Dr. Ganger also notes that a file system could have a set of predetermined locations that might hold the root inode. However, Dr. Ganger has not demonstrated how the specification might contemplate such an embodiment, and the specification itself repeatedly refers to a "fixed location," as discussed above. However, if the set were small enough that the root inode could be readily located, it might constitute a fixed location (or at least its equivalent), as Sun's expert acknowledged at the hearing.

9/10/08 Order at 44. However, this observation was made in a different context. Specifically, the experts at the claim construction hearing were discussing copies of the root inodes that are always simultaneously located at a few (i.e., two or three) fixed locations. Williamson Decl., Ex. 8 (Hearing Tr.) at 83-84 (after Dr. Brandt described a set of three fixed locations, Dr. Ganger noted that the patent only describes two fixed locations at which the root inode is always present). In contrast, here the uberblock is not always located in a small set of two or three locations, but its location is constantly changing. The Court also noted at claim construction that the patent did not contemplate a set of predetermined locations. 9/10/08 Order at 43-44. It certainly does not contemplate that a set as large as those discussed here (128, or 512, or even thousands – as opposed to 2 or 3) could constitute a fixed location simply by virtue of being contained within a larger fixed entity (such as the vdev label).

In support of its "small set of locations" argument, NetApp also asserts that Dr. McKusick confirmed that a set of locations of less than four that might hold the root inode might constitute a fixed location or at least its equivalent. Homrig Decl., Ex. 1 at 297-299. NetApp therefore argues that there is a factual dispute over what number of possible locations would be sufficiently small

such that the root inode could be in a fixed location. While Dr. McKusick suggests that the number is "less than four," NetApp says this opinion is subject to reasonable differences of opinion and depends on context, and that given the operating speed for computers today, a reasonable juror could conclude that 512 possible locations is sufficiently small, especially relative to the overall number of possible locations on a disk. NetApp also relies on Mr. Bonwick's testimony to argue that an uberblock stored in one of 128 possible 1KB locations requires looking at a "relatively small set of data." Ex. 15 (Bonwick Depo.) at 251-52 ("uberblocks are part of the relatively small set of data in ZFS that it is [sic] at a particular location on disk. . . you have to start somewhere. So there has to be something that's at a known location on disk"). However, Mr. Bonwick was referring to the 128 KB size of each of the uberblock arrays in the vdev labels, which ignores the fact that in a single disk system, 512 locations must be searched, and in a multi-disk system, up to thousands of locations must be searched.

Finally, NetApp equates a "fixed" location with a "known" location, and this interpretation is supported to some extent by Sun's own witness. Dr. McKusick stated that if something is in a known location, it is in a fixed location. Homrig Decl., Ex. 1 (McKusick Depo) at 39. He also stated that all four vdev labels are in fixed locations because they are known locations. Id. This is true even though some run time computation is required to locate the back two vdev labels. Id. at 42. NetApp also points out that Dr. McKusick described something as a fixed location in the Rosenblum dissertation, even though the block number is calculated at run time. Therefore, NetApp argues that there is evidence in the record that something may be fixed yet still involved certain run time location to locate it. See Homrig Decl., Ex. 35 (McKusick Expert Report) at ¶ 37 (describing checkpoint region in Rosenblum dissertation as fixed location despite fact that Rosenblum dissertation (Id., Ex. 36 at 49) states that the checkpoint region location is stored in a superblock, so that block number is calculated at run time)). However, this focus on "run time" misses the point. The issue is not how long it takes to find a given location, but whether the location is known from the beginning of the search (regardless of how long it takes to reach the location). Were the Court to rely on "run time" as a factor in determining whether something is in a fixed location, this would mean that the likely inevitable speed-up in run time technology, unrelated to the patent at issue or to

the allegedly infringing invention, could turn technology that had for a long time been non-infringing into an infringing invention.[2]

For all of the foregoing reasons, the Court holds that "fixed" is not so broad as to encompass the hundreds of possible locations in which the ZFS uberblock might reside.

D. No Reasonable Juror Could Find That The Limitation of A "Fixed" Location Is Met When Comparing The Term As Construed To the ZFS Technology

In addition to the foregoing, NetApp argues that some evidence in the record indicates that the uberblock itself is in a fixed location on disk. See Homrig Decl., Ex. 37 (Bonwick email) at SUN000204104 (uberblock gets "rewritten in a fixed location at the end of each transaction group"); Ex. 29 (Moore Depo.) at 160 ("You could start from some block in a relatively well known location, the uberblock in ZFS's case"); Ex. 30 (Maybee Depo.) at 106-07, 184 (uberblock is "within a fixed area of the vdev label" and the uberblock is the only thing that is fixed in a given scenario). However, Mr. Bonwick's testimony instead supports Sun's position, in that it shows that after each transaction the uberblock gets rewritten to a different location (and is therefore necessarily not fixed). Sun Reply at 4-5. As to Mr. Maybee's testimony that the uberblock is within a fixed area of the vdev label, Sun points out that Mr. Maybee actually stated that the uberblock can appear anywhere within that portion of the vdev label (the uberblock array), and that the uberblock is within that fixed area of the vdev label, which is not in dispute. See Homrig Decl., Ex. 30 at 106-07. Instead, Mr. Maybee's testimony indicates that a new uberblock is written to a new location after every transaction (in a round robin fashion), and such a process does not result in a "fixed" location as defined above. See Williamson Decl., Ex. 6 at 195-96, 201. Mr. Maybee did also state that the uberblock is in a "relatively well known location," id., Ex. 30 at 160, and an email in the record

---

[2]NetApp also argues that the amount of space allocated in ZFS for the uberblock relative to the rest of the space on a typical disk is comparable to the amount of space in WAFL allocated to storing the root inode. See Homrig Decl., Ex. 1 (McKusick Depo.) at 143-144. However, Sun points out that Dr. McKusick also testified that: (1) in 1994, a system only would need to read 4KB of data, whereas today a system using ZFS would need to read a megabyte of data, (2) all four vdev labels on a ZFS system must be read as opposed to the single fsinfo structure, (3) it takes longer to read one megabyte off of a modern disk than it took to read 4 kilobytes off of a 1994 disk, (4) the amount of data read in a ZFS system is greater in a multi-disk system. Homrig Decl., Ex. 1 at 143-44. Thus, NetApp's evidence does not support its argument that the time or method for searching for uberblocks in ZFS is minimal/comparable to the process in WAFL for locating the root inode. Further, the issue before the Court relates to the number of locations that need to be searched, not the time required for the search.

states that "the vdev label (including the uberblock) has a fixed location, everything is derived from that." See Homrig Decl., Ex. 34. However, these statements do not focus on the distinction between the fixed nature of the vdev labels and uberblock arrays, as opposed to the location of the active uberblock itself, and "relatively" is a very vague term, so this evidence does not raise a triable issue of fact.

With respect to a multi-disk system, Sun also argues that if the hardware product that is running ZFS has more than three vdevs in a pool, the active uberblock is stored on a different set of three randomly selected vdevs each time a new uberblock is written. McKusick Decl., ¶ 11; Bonwick Decl. ¶ 6. Therefore, the active uberblock may not even be on a fixed disk within the pool. Because the system must search all of the disks in a pool to find the active uberblock, the system must scan possible locations spread over multiple disks in the pool in which the active uberblock may be located. McKusick Decl. ¶ 13; Bonwick Decl. ¶ 8. Sun's argument is essentially that no reasonable jury could find that an uberblock that moves among 128 possible locations in an array, and that also moves among different sets of disk in the pool, which requires the system to search up to thousands of disk locations to find the active uberblock, resides in a fixed location.

NetApp counters that sometimes copies of the active uberblock are not only written to all four vdev labels on one disk, but to multiple disks in multiple disk configurations. NetApp Opp. at 17:1-4. NetApp appears to be arguing that in a multi-disk system, because there are many copies, not as many locations need be searched as it may appear. But in this scenario, the vdevs to which the uberblock are written are randomly selected each time a new active uberblock is written, and at any given time, the system does not know precisely where the active uberblock is located. See Homrig Decl., Ex. 1 (McKusick Depo) at 264-65 ("if you had – if one of those VDEVs, for example, was RAID and had six drives in it, when we talk about writing to that – let's say we had five of them and say we had three of them that we had selected to write to uberblock 2, we would actually end up writing to 18 drives. So it would be all six of each of the three logical VDEVs that would have been selected and all four labels on all of those VDEVs"). Therefore, the Court agrees with Sun's position.

E. ZFS's Uberblock Does Not Satisfy The On-Disk Root Inode Claim Under The Doctrine of Equivalents

In applying the doctrine of equivalents, the "all elements rule" requires that equivalence be addressed on a limitation-by-limitation basis, rather than from the perspective of the invention as a whole, and that no limitation be read completely out of the claim. Freedman Seating Co. v. American Seating Co., 420 F.3d 1350, 1358 (Fed. Cir. 2005); Tronzo v. Biomet, Inc., 156 F.3d 1154, 1160 (Fed. Cir. 2998) (holding that the patentee's theory of equivalence – that any shape would be equivalent to the conical limitation – would write that limitation out of the claims). However, a claim element is not vitiated merely because it does not literally exist in the accused product. Rather, "[a] holding that the doctrine of equivalents cannot be applied to an accused device because it 'vitiates' a claim limitation is nothing more than a conclusion that the evidence is such that no reasonable jury could conclude that an element of an accused device is equivalent to an element called for in the claim, or that the theory of equivalence to support the conclusion of infringement otherwise lacks legal sufficiency." DePuy Spine, Inc. v. Medtronic Sofamor Danek, Inc., 469 F.3d 1005, 1018-19 (Fed. Cir. 2006) (citation omitted).

In this case, the doctrine of equivalents is not met. First, instead of storing an on-disk root inode at a fixed location on disk so that the system knows where it is and can immediately locate it as taught by the '211 patent, ZFS was specifically designed to constantly change the uberblock locations both on and between disks. This design achieved two advantages: (1) to permit a system that runs ZFS to use storage media that are, unlike hard disk drives, susceptible to being worn out if the same location on the media always is used to store an uberblock, thereby permitting the use of flash memory, and (2) to not overwrite the active uberblock. Sun made this design choice because it concluded that the additional computational time required to find the uberblock was worthwhile to achieve these goals. McKusick Decl. ¶¶ 9, 16-17; Bonwick Decl. ¶ 9. In support of its doctrine of equivalents argument, Sun also contends that whether something is in a fixed location or not is a requirement that is binary in nature, and that NetApp has not explained where the range of equivalents begins and ends under its theory. Sun analogizes the case to Freedman Seating Co. v. American Seating Co., 420 F.3d 1350 (Fed. Cir. 2005), in which the Federal Circuit rejected the

argument that a support member that was "confined to a fixed location" was the equivalent of the claimed "slidably mounted" support member, because to do so would vitiate the claim limitation. Id. at 1361-62. NetApp counters that there is a factual dispute as to whether a set of 128 predetermined locations for the active uberblock is a substantially similar way of ensuring that the root of the file system can be found, relying again on the Court's prior statement that if the set were small enough that the root inode could be readily located, it might constitute a fixed location or its equivalent. See 9/10/09 Order at 44.[3]

The Court would not go so far as Sun does in arguing that the fixed location is entirely binary in nature, as both the Court and the experts have recognized that some small set of locations (i.e., two, or maybe four) could be seen as equivalent to fixed. However, nothing in the record indicates that the uberblock, which can be located in a minimum of 512, or even thousands, of locations is equivalent to the '211 patent's "fixed location" requirement. No reasonable jury could conclude that this aspect of the uberblock is equivalent to the fixed element in the claim. See Sage Prods. Inc. v. Devon Indus., Inc., 126 F.3d 1420, 1423 (Fed. Cir. 1997) (though equivalence is a factual matter normally reserved for a factfinder, court may grant summary judgment where it concludes that no reasonable jury could determine two elements to be equivalent); see also id. at 1425 ("Because this issued patent contains clear structural limitations, the public has a right to rely on those limits in conducting its business activities.").

In addition to vitiating the claim limitation, NetApp has not shown that either the insubstantial differences test or the function-way-result tests are met. In applying the "insubstantial differences" test, Sun's explanation of the intentional design choices behind Sun's decision to have the uberblock change locations on and between disks (to permit a ZFS system to use flash memory and not overwrite the active uberblock) is persuasive. These two functions are not functions of the

[3]NetApp also notes that because its WAFL technology is a "pioneer invention," it is "entitled to a broad range of equivalents." See Perkin-Elmer Corp. v. Westinghouse Elec. Corp., 822 F.2d 1528, 1532 (Fed. Cir. 1987); Regents of the Univ. of Cal. v. Dako N. Am., Inc., 2009 WL 1083446 at *11 (N.D. Cal. 2009) (finding that the value of the patent system would be diminished if slight improvements upon fundamental biotechnology methods could escape infringement of a patent to a pioneer invention). Sun does not appear to contest that the invention is a pioneer invention, though neither party has analyzed the question in detail. However, even affording NetApp the leeway that may be given to pioneer patents, in viewing the stark differences between the device and the patent, no reasonable factfinder could find the doctrine of equivalents to be met in this case.

claimed on-disk root inode, and the way the active uberblock operates – by moving on and among different disks – is the antithesis of an inode in a known, fixed location on disk. The intentional differences between the accused system and the '211 patent are substantial, because instead of storing an on-disk root inode at a fixed location on disk so that the system knows where it is, ZFS was specifically designed to store the uberblock in a constantly changing location both on and between disks. This preclude a finding of infringement under the doctrine of equivalents. See Honeywell Int'l Inc. v. Hamilton Sundstrand Corp., 370 F.3d 1131, 1139 (Fed. Cir. 2004) ([a]n element in the accused device is equivalent to a claim limitation if the only differences between the two are insubstantial"); Warner-Jenkinson, 520 U.S. at 39 n.8; Sage Prods. Inc. v. Devon Indus. Inc., 126 F.3d 1420, 1424 (Fed. Cir. 1997) (noting that the "doctrine of equivalents prevents an accused infringer from avoiding infringement by changing only minor or insubstantial details of a claimed invention while retaining their essential functionality").

Applying the function-way-result test, Sun also correctly points out that the two important functions and corresponding result of having a moving uberblock discussed above (to permit a ZFS system to use flash memory and not overwrite the active uberblock) are neither the functions nor the result of the claimed on-disk root inode. In Crown Packaging Tech., Inc. v. Rexam Bev. Can Co., 559 F.3d 1308, 1315 (Fed. Cir. 2009), the Court held that summary judgment was improper where expert testimony conflicted as to the function of the invention. Here, however, NetApp does not dispute this point regarding functions, and Dr. Ganger does not address these functions in his expert report. See Ganger Report ¶ 42. Therefore, this case is distinguishable from Crown Packaging and summary judgment is appropriate for this reason as well.

Moreover, the way the active uberblock operates by constantly moving is the antithesis of an inode that is a known, fixed location on disk or in two concurrent fixed locations in dependent claim 4 of the patent (the way of the '211 patent). NetApp argues that there is a factual dispute as to whether a set of 128 predetermined locations for the active uberblock is a substantially similar way of ensuring the root of the file system can be found, relying again on the Court's prior statement that if the set were small enough that the root inode could be readily located, it might constitute a fixed location or its equivalent. 9/10/09 Order at 44. But NetApp fails to explain how such a large set of

locations (which as discussed above, is a minimum of 512 and can be in the thousands) can involve the same "way" of having a root inode in a fixed location on disk. Dr. Ganger did opine that "having uberblock arrays in vdev labels operates in substantially the same way as a fixed root inode, because both reliably and efficiently use a fixed location to facilitate finding the root inode during booting." Ganger Report ¶ 42. However, Dr. Ganger's statement is not specific to the uberblock and does not explain how these functions operate in the same way, and is therefore too conclusory to create a triable issue of fact. See Dynacore Holdings Corp. v. U.S. Philips Corp., 363 F.3d 1263, 1278 (Fed. Cir. 2004) ("expert's unsupported conclusion on the ultimate issue of infringement is insufficient to raise a genuine issue of material fact").

E. Sun's Additional Summary Judgment Requests

In footnote 5, Sun requests that the Court find that it is impossible for Solaris and ZFS to directly infringe any claim of the '211 patent on the ground that as software they do not include any of the hardware claim limitations of the independent claims. However, Sun offers no evidence in support of this argument. In footnote 6, Sun asks that the Court rule that its separate distribution to third parties of Solaris 10 OS not packaged with a Sun hardware product does not contributorily infringe the patent, because it has an admittedly substantial noninfringing use – the use of UFS rather than ZFS. These arguments were not adequately briefed, could be seen as improper attempts at additional summary judgment motions, and were abandoned on Reply. Therefore the Court need not address them.

For all of the reasons stated above, summary judgment is granted as to non-infringement of NetApp's '211 patent.

IT IS SO ORDERED.

Dated: November 16, 2009

ELIZABETH D. LAPORTE
United States Magistrate Judge

UNITED STATES DISTRICT COURT

FOR THE

NORTHERN DISTRICT OF CALIFORNIA

NETWORK APPLIANCE INC,

Plaintiff,

v.

SUN MICROSYSTEMS INC et al,

Defendant.
_______________________________/

Case Number: CV07-06053 EDL

CERTIFICATE OF SERVICE

I, the undersigned, hereby certify that I am an employee in the Office of the Clerk, U.S. District Court, Northern District of California.

That on November 17, 2009, I SERVED a true and correct copy(ies) of the attached via email to:

Counsel for Sun Microsystems: Christine Kerba Corbett
DLA Piper US LLP
christine.corbett@dlapiper.com

Counsel for Network Appliance: Edward Robert Reines
Weil Gotshal & Manges LLP
Edward.Reines@weil.com

Dated: November 17, 2009

Richard W. Wieking, Clerk

Lili M Harrell

By: Lili M. Harrell, Deputy Clerk